Kubernetes-based MEC Architecture

Van-Binh Duong, Minh-Ngoc Tran and Younghan Kim*
Soongsil University

binhdv@dcn.ssu.ac.kr, mipearlska1307@dcn.ssu.ac.kr, younghak@ssu.ac.kr *

Abstract

Multi-access Edge Computing has brought great benefits to the development of networks. It gives the network the ability to provide storage and compute capabilities at the edge, which helps reduce latency, improve data privacy, and fulfill the requirements of many increasing users. However, the current design of MEC architecture needs to keep up with the updates of technologies, especially in the cloud, regarding network characteristics, management, and environments. Meanwhile, Kubernetes is currently well-known for its accelerations in scalability and management of clusters. Thus, it is necessary to provide a straightforward and flexible design that can effortlessly adapt MEC to Kubernetes to take advantage of the efficiency of Kubernetes in cloud environments. Consequently, the research aims to contribute to the design of MEC orchestration targeting cloud environments. The architecture evaluations show that the proposed design provides high capabilities of scalability and computing management.

I. INTRODUCTION

The introduction of the 5G network has made a large improvement in internet connection speed affecting not only life living but also industrial developments. This leads to the rise of security and scalability problems caused by the increase of provided services and users. Therefore, MEC has been introduced as a technology to enable the network to provide computing and storage capabilities at the edge in the purpose of giving near real-time services and securing the user data. Nevertheless, developing an effective MEC orchestration is challenging with many factors that need to be considered (e.g., networking, technology, system adaptation, etc.). As introduced in the ETSI GR MEC 017 [1], the integration of Network Function Virtualization (NFV) and MEC placed a first step to adopting MEC into cloud. However, there are overlaps and partial compatibilities in MEC and NFV components which could prevent MEC from providing the best performance. As a result, the paper proposes a design of Kubernetes-based MEC orchestration architecture for cloud. With benefits from Kubernetes, MEC architecture is redesigned to give more flexibility in both management and scalability.

II. BACKGROUND

A. MULTI-ACCESS EDGE COMPUTING

Multi-access Edge Computing (MEC) [2] is the implementation of mobile edge applications as software-only entities that run on top of a virtualization infrastructure, which is located in or close to the network edge. The Mobile Edge Computing framework shows the general entities involved. MEC architecture comprises Mobile edge system level, Mobile edge host level, and Mobile edge network level entities. The mobile edge host level is constructed by two components Mobile edge host and Mobile edge host level management.

B. MEC IN NFV ENVIRONMENT

When integrating MEC into NFV [1], MEC instantiates Virtualized Network Functions (VNFs) near end users by deploying computing resources at the edge through the usage of NFV's virtualization concepts. The MEC platform is deployed as a VNF using the ETSI NFV protocols. Reusing existing ETSI NFV MANO functionality is possible via MEC apps. The Virtualization infrastructure is deployed as a Network Functions Virtualization Infrastructure (NFVI), and its virtualized resources are managed by the ETSI NFV-defined Virtualized Infrastructure Manager (VIM). The notions of NFVI-PoP (NFVI Point of Presence) replace the traditional idea of a MEC host in this context. The Mobile Edge Platform Manager-NFV (MEPM-V), which replaced the MEC platform manager, assigns one or more Virtual Network Function Managers (VNFM) to handle the Life-cycle management (LCM) component. However, the Mobile Edge Application Orchestrator (MEAO) has replaced the previous name of the MEC orchestrator. The MEAO interfaces with the NFV Orchestrator (NFVO) to utilize its resource orchestration and the orchestration of the set of MEC application VNFs as one or more NFV Network Services (NSs). However, the challenging issue is the mapping of the concept of MEC host to NFV. Both NFV and MEC use descriptors to define information to instantiate a VNF and an Application, respectively. It causes the descriptors partial compatibility issue in this design.

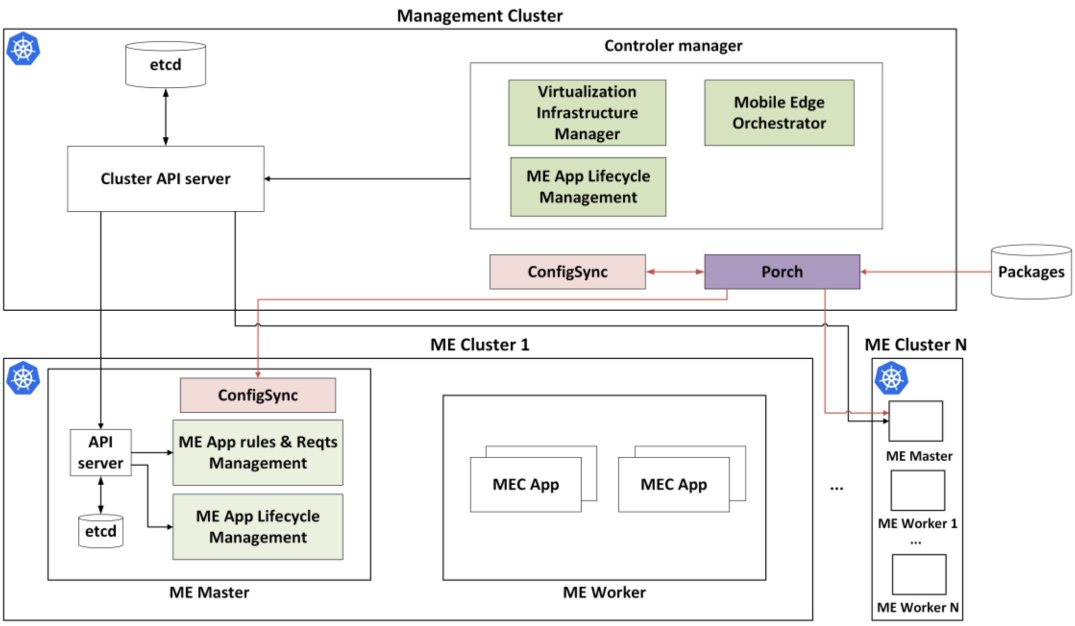

Figure 1: Kubernetes-based MEC architecture.

III. PROPOSED ARCHITECTURE

To obtain a simple and flexible architecture design of MEC for service orchestration in cloud, this paper proposes the use of Kubernetes in designing MEC. In the Kubernetes (K8s) environment, MEC concepts can be replaced with available ones in K8s and supported technologies in K8s ecosystem and some entities in MEC can be reduced. As in Figure 1, the MEC DNS handling is not shown because DNS is taken care of by Kubernetes' default components. The strength of the Kubernetes-based design is the scalability of the system where many clusters can be deployed and managed by a central management cluster.

Furthermore, management entities of MEC could be developed as Custom Resource Definitions (CRD) which is a technique to extend K8s functions. Figure 1 shows that components such as Mobile Edge Orchestrator are in the management cluster to handle the management flows. They could be CRDs managing sub-CRDs in clusters and, in clusters (e.g., Cluster 1 in Figure 1), management components located in ME Master are sub-CRDs using CR provided by controller CRDs to manage corresponding working flows inside the cluster.

Concretely, there are two sorts of MEC management components, the first one includes Mobile Edge Orchestrator (MEO), Virtualization Infrastructure Manager (VIM), and ME App Lifecycle Management (MEALM) which are located in the management cluster. The MEO is responsible for deploying ME clusters, the resource availability in each cluster, and the applications that are instantiated. The VIM's role is to manage the underlying virtualized infrastructure that supports the cloud services, monitor the performance and health of virtualized resources, and generate related reports. The MEALM involves the instantiation and termination of the applications and applies traffic rules and requests to deployed applications in the cluster. Meanwhile, the second sort comprises ME App rules & Reqts Management and ME App Lifecycle Management. They are located in the ME Master of each ME cluster to handle management requests from MEALM.

Besides, ConfigSync [3] and Porch [4] provide the GitOps and configuration automation ability for the K8s cluster. ConfigSync allows operators to manage K8s deployments. It can support managing multi-clusters, role bindings, resource quotas, etc. Porch is used with ConfigSync to automate the configuration of interconnected network functions and the underlying infrastructure. It also supports the revisions of deployed clusters. This means the older cluster configurations can be saved as packages and re-deployed when needed.

IV. CONCLUSION

In this paper, a proposal of Kubernetes-MEC deployment architecture is introduced. With the advantages of Kubernetes, most aforementioned problems of NFV-MEC can be resolved. Kubernetes provides MEC with an extreme capability to keep up with updates of cloud techniques as well.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2020-0-00946,Development of Fast and Automatic Service recovery and Transition software in Hybrid Cloud Environment).

REFERENCES

[1] ETSI GR MEC 017 V1.1.1, "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment", 2018.
[2] ETSI GS MEC 003 V1.1.1, "Mobile Edge Computing (MEC); Framework and Reference Architecture", 2016.
[3] "Config Sync Overview | Config & Policy." Google Cloud, Accessed 8 Jan. 2024.
[4] "Package Orchestration Server." Kpt Documents. Accessed 8 Jan. 2024.